15-23, 25-36 and claims 38-46, which depend from claims 1, 14, 24 and 37, respectively, were also rejected under 35 USC §103(a) as being unpatentable over Madukkarumukumana in combination with a number of references, including VIA. Furthermore, claims 8-13 and 31-36 were provisionally rejected under the judicially created doctrine of obviousness-type double patenting over copending Application No. 09/458,240 in view of Madukkarumukumana and VIA and claims 1-23 and 24-46 were also provisionally rejected under the judicially created doctrine of obviousness-type double patenting over copending Application No. 09/458,138.

To overcome the obviousness-type double patenting rejections, Applicants attach to this response a terminal disclaimer, which overcomes the provisional rejections.

The Office Action dated October 30, 2002 has been carefully considered. Applicants would like to thank Examiner Zhen for the courtesy of a telephonic interview on January 21, 2003 regarding the Office Action in co-pending Application No. 09/458,138, which is related to the present application. During that interview, Examiner Zhen indicated that the Madukkarumukumana reference, used in both that Office Action and the Office Action in the present case, described the replacement of Remote Procedure Call ("RPC") mechanisms with a custom marshalling. Specifically, the Examiner pointed to Figure 5, illustrating the replacement of "Standard RPC" with a "VIA Transport" through a "Custom Proxy Manager" and a "Custom Stub". *See* Madukkarumukumana, Figure 5. The Examiner also referenced section 4.2, which states "each interface stub unmarshals method parameters from receive buffers, dispatches actual object methods, marshals return parameters into the reply buffers, and returns to the stub manager", further suggesting that the RPC layer, which is responsible for those functions, was bypassed. *See* Id., p. 5, sec. 4.2, lines 12-16.

Applicants agree with the Examiner that the Madukkarumukumana reference does teach the replacement of RPC mechanisms with a custom marshalling. The present application, on the other hand, does not replace the RPC mechanisms. To the contrary, the independent claims of the present application specifically require that the claimed "first computer" or the claimed "second computer" have "a Remote Procedure Call layer, wherein the RPC layer has access to an RPC buffer". Furthermore, each of the independent claims recites specific actions to be performed by the RPC layer. Consequently, as will be described in greater detail below, the present application describes and claims a system that is far different than that taught by Madukkarumukumana.

Generally, the present application is directed to increasing the efficiency of communications between two objects residing on two computing devices. One mechanism by which the efficiency of these communications can be increased is by eliminating copying steps performed when parameters or data are marshaled between the two objects. Specifically, rather than copying the parameters or data into buffers, from which the networking hardware can read and transmit them, the present application describes a mechanism for providing only a list of pointers to memory locations containing those parameters or data, avoiding the need to copy the parameters or data. *See e.g.,* Application, p. 18, line 16 – p. 19, line 13. However, it remains desirable to utilize the mechanisms of the RPC layer while avoiding copying steps. *See e.g.,* Id. Therefore, to more precisely claim this aspect of the present application, independent claims 1 and 24 have been amended to now recite a "first computer" having "a Remote Procedure Call layer, wherein the RPC layer has access to an RPC buffer" while independent claims 14 and 37 have been amended to now recite a "second computer" having "a Remote Procedure Call layer, wherein the RPC layer has access to an RPC buffer".

As explained in both the present application and in the Madukkarumukumana reference, the Distributed Component Object Model ("DCOM") is built on top of RPC, and avails itself of RPC mechanisms contained in the RPC layer. *See generally*, Id., p. 23, lines 8-9; Madukkarumukumana, p. 3, sec. 3.1, lines 5-7. However, when a custom marshalling is used, all of the underlying RPC mechanisms are bypassed by the custom marshalling. For example, the RPC transport layer is bypassed. *See* Madukkarumukumana, p. 2, sec. 1, lines 1-7 ("this paper provides the following two contributions :... : A specialized methodology to replace legacy RPC transports in DCOM"). Any security provided by RPC is also bypassed. *See* Id., p. 4, sec. 4.1, lines 25-27 ("security features are not present in the current custom marshaler") Thus, when the Madukkarumukumana reference describes a custom marshalling scheme, that custom marshalling scheme does not use any of the mechanisms of the RPC layer. *See generally*, Madukkarumukumana, Figure 5.

The Office Action relies on Madukkarumukumana to reject all of the independent claims, specifically claims 1, 14, 24 and 37. However, all of the independent claims require the existence and use of the RPC layer. Thus, the present application describes and claims the use of the very RPC functions which are bypassed by the custom marshalling of Madukkarumukumana.

Specifically, the present application describes the use of "the RPC layers to communicate the call parameters across the network". Application, p. 18, lines 21-22. For example, "the RPC run-time layers 126 and 136 interpret the data 152 and 156 as a list of scatter-gather entries, each comprising a starting memory address of the data they point to and the length of the data. As shown in Figure 4A, the RPC run-time layer 126 adds RPC headers to the list 152 and passes it to the loadable transport layer 128." Id. at p. 19, lines 4-8. Other RPC mechanisms, such as security, thread management, and socket connection management, are also utilized. *See* Id., p.

25, lines 15-16. Clearly, the present application does not describe the bypassing of the RPC mechanisms and, indeed, the claims specifically require the performance of functions by the RPC layer. For example, independent claims 1 and 24 require that the RPC layer treat "the first pointer as a scatter-gather entry" while the remaining independent claims, claims 14 and 37, require the RPC layer to store "the parameter in a memory location". This is the opposite of what is described in Madukkarumukumana, where the RPC layer is completely bypassed, resulting in a situation in which "security features are not present in the current custom marshaler" and "a specialized methodology … replace[s] legacy RPC transports." *See* Madukkarumukumana, p. 4, sec. 4.1, lines 25-27; p. 2, sec. 1, lines 1-7. Consequently, because Madukkarumukumana describes a custom marshalling which bypasses all of the RPC mechanisms, while all of the independent claims of the present application claim the use of the RPC layer, it is respectfully submitted that the pending claims are allowable over the prior art of record.

In addition to the above described claim amendments, claims 8, 11, 19, 31, 34 and 42 were amended to recite at least two buffers: a send buffer and a receive buffer, and specify that the receive buffer is selected with the intent that it be sufficiently large to accept the incoming data. The Office Action rejected these claims based on combinations involving Madukkarumukumana, U.S. Patent No. 6,044,409 to Lim et al. ("Lim"), and U.S. Patent No. 6,131,126 to Kougiouris et al. ("Kougiouris"). However, the Lim reference never teaches or suggests that the size of the buffer is in any way affected by the possible size of the response data. Rather, the size of the buffer used by Lim is exclusively determined by the size of the data being sent. The present application, however, describes the pre-posting of a receive buffer, which is not used to send data, and whose size is selected so that it should be able to store the data that will be received. The size of the receive buffer is, therefore, affected by the possible size of the data that will be received. Additionally, the Kougiouris reference never teaches or suggests the maintenance of buffers for a sufficient period of time before they are cleared. Instead, Kougiouris describes the allocation of memory space in an operating system on which both communicating applications are running, such that the operating system accepts data from one and then transfers it to the other. Because the sending program can rely on the operating system to negotiate the transfer of data to the destination program, it can clear its own buffers whenever it wishes. However, the communicating programs described by the present application do not share a common operating system and, therefore, must each maintain the data in their own buffers until it has been properly received by the destination program.

As has been shown, independent claims 1, 14, 24 and 37 are not rendered obvious by the Madukkarumukumana reference in combination with VIA or additionally, in combination with the additional references cited, since Madukkarumukumana specifically teaches against the use of the RPC layer. Furthermore, because the remaining claims are dependent from claims 1, 14, 24 or 37, they are allowable for the same reasons. Finally, as shown above, dependent claims 8, 11, 19, 31, 34 and 42 are additionally allowable because neither the Lim nor Kougiouris references teach or suggest all of the elements of those claims, as amended. As a result, it is respectfully submitted that all of the claims, as amended, are allowable over the prior art of record.

CONCLUSION

In view of the above amendments and remarks, the application is considered in good and proper form for allowance, and the Examiner is respectfully requested to pass this application to issue.

If, in the opinion of the Examiner, another telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney.

Respectfully submitted,

Vladan M. Vasiljevic, Reg. No. 45,177
One of the Attorneys for Applicant(s)
LEYDIG, VOIT & MAYER, LTD.
Two Prudential Plaza, Suite 4900
180 North Stetson
Chicago, Illinois 60601-6780
(312) 616-5600 (telephone)
(312) 616-5700 (facsimile)

Date: February 27, 2003